Zadanie 6

Treść

Ułóż algorytm, który dla danego n-wierzchołkowego drzewa i liczby k, pokoloruje jak najwięcej wierzchołków tak, by na każdej ścieżce prostej było nie więcej niż k pokolorowanych wierzchołków.

Algorytm

Będziemy iteracyjnie kolorować warstwy liści w drzewie.

1. Zainicjuj zbiór pokolorowanych wierzchołków $C := \emptyset$.
2. Zainicjuj roboczą kopię drzewa $T' := T$.
3. Wykonaj $\lfloor \frac{k}{2} \rfloor$ razy:
   i. Jeśli T' jest pusty, zakończ pętlę.
   ii. Znajdź zbiór L wszystkich liści w aktualnym drzewie T'.
   iii. Dodaj wszystkie wierzchołki z L do zbioru C.
   iv. Zaktualizuj T' poprzez usunięcie z niego wszystkich wierzchołków ze zbioru L.
4. Jeśli k jest nieparzyste i graf T' wciąż zawiera jakieś wierzchołki:
   i. Wybierz dowolny wierzchołek v należący do T'.
   ii. Dodaj v do zbioru C.
5. Zwróć zbiór C.

Dowód poprawności

Musimy pokazać, że na dowolnej ścieżce w oryginalnym drzewie T znajduje się co najwyżej k pokolorowanych wierzchołków.

Rozważmy dowolną ścieżkę P w drzewie T. Ścieżka ta może wejść do pewnej warstwy i ją opuścić co najwyżej raz. Oznacza to, że P może zawierać co najwyżej dwa wierzchołki z dowolnej warstwy.

1. ***k* jest parzyste**:

   Algorytm wykonuję pętlę $\lfloor \frac{k}{2} \rfloor = \frac{k}{2}$ razy. Oznacza to, że koloruje dokładnie $\frac{k}{2}$ najbardziej zewnętrznych liści. Każda z tych warstw może dodać do naszej ścieżki co najwyżej 2 pokolorowane wierzchołki. W najgorszym przypadki ścieżka przejdzie przez każdą warstwę 2 razy. Całkowita liczba wierzchołków na ścieżce nie przekroczy więc k.

2. ***k* jest nieparzyste**:

   Algorytm wykonuje pętlę $\lfloor \frac{k}{2} \rfloor = \frac{k-1}{2}$ razy. Koloruje więc $\frac{k-1}{2}$ warstw. Te warstwy mogą dać co najwyżej $k-1$ pokolorowanych wierzchołków na dowolnej ścieżce, więc po pokolorowaniu jednego dodatkowego wierzchołka, żadna ścieżka nie będzie miała więcej niż k pokolorowanych wierzchołków.

Złożoność

Na początek musimy znaleźć wszystkie liście, musimy przejść po każdym wierzchołku zatem zajmie to $O(n)$. Podczas usuwania wierzchołków w aktualnym L możemy tworzyć kolejne L', które będzie zawierało liście które się utworzą. Takich operacji znowu wykonamy $O(n)$. Wszystkie inne operacje powinny nas kosztować stały czas.

Wykonanie: $O(n)$; Złożoność pamięciowa: $O(n)$.

Opracował: Filip Figzał, 2025

Maksymalność

Dowód przeprowadzimy za pomocą argumentu wymiany. Pokażemy, że dowolne optymalne rozwiązanie C_{OPT} można krok po kroku przekształcić w rozwiązanie wygenerowane przez nasz algorytm, C_{ALG}, nie zmniejszając przy tym liczby pokolorowanych wierzchołków.

Załóżmy, że $k \geq 2$, ponieważ dla $k = 0, 1$ możemy pokolorować T w trywialny sposób.

Niech C_{OPT} będzie optymalnym kolorowaniem. Chcemy pokolorować wszystkie liście L_0 w drzewie T. Jeśli $L_0 \subseteq C_{\text{OPT}}$ (wszystkie liście są pokolorowane), to C_{OPT} zgadza się z C_{ALG} na pierwszej warstwie i możemy kontynuować rekurencyjnie dla drzewa $T - L_0$ i parametru $k - 2$.

Załóżmy, więc że istnieje liść $v \in L_0$, który nie jest pokolorowany w rozwiązaniu optymalnym, czyli $v \notin C_{\text{OPT}}$. Pokażemy, że możemy zmodyfikować C_{OPT} tak, aby zawierało v, nie tracąc na optymalności.

Rozważmy nowe kolorowanie $C_{\text{test}} = C_{\text{OPT}} \cup \{v\}$.

1. **Kolorowanie C_{test} jest poprawne**:

   Jeśli C_{test} jest poprawne, to znaleźliśmy rozwiązanie lepsze od optymalnego, ponieważ $|C_{\text{test}}| = |C_{\text{OPT}}| + 1$. To jest sprzeczność z założeniem, że C_{OPT} jest optymalne. Ten przypadek nie może więc zajść. ↯

2. **Kolorowanie C_{test} jest niepoprawne**:

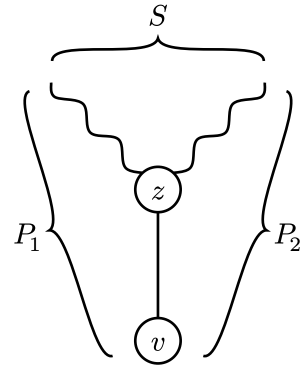
   Skoro C_{test} jest niepoprawne, to musi istnieć co najmniej jedna ścieżka, na której znajduje się $k + 1$ pokolorowanych wierzchołków. Nazwijmy zbiór takich „zepsutych" ścieżek $\mathcal{P}_{\text{zepsute}}$.

   Każda ścieżka $P \in \mathcal{P}_{\text{zepsute}}$ musi spełniać dwa warunki

   i. $v \in P$, ponieważ bez v kolorowanie było poprawne.
   ii. $|P \cap C_{\text{OPT}}| = k$, czyli przed pokolorowaniem v, ścieżka miała k pokolorowanych wierzchołków.

   Lemat 1 *Istnieje pokolorowany wierzchołek w wspólny dla wszystkich zepsutych ścieżek.*

   Niech $P_1, P_2 \in \mathcal{P}_{\text{zepsute}}$ będą dwiema różnymi zepsutymi ścieżkami. Jako że obie zaczynają się w liściu v, muszą posiadać pewną część wspólną. Niech z będzie wierzchołkiem, w którym te ścieżki się rozchodzą.



Rysunek 1: Poglądowy rysunek opisanej sytuacji

   Załóżmy, dla dowodu nie wprost, że na wspólnej części ścieżki od v do z nie ma żadnego wierzchołka z C_{OPT}. Oznacza to, że wszystkie k pokolorowane wierzchołki z $P_1 \cap C_{\text{OPT}}$ leżą na gałęzi za z, a wszystkie k pokolorowane wierzchołki z $P_2 \cap C_{\text{OPT}}$ leżą na swojej gałęzi za z.

Rozważmy teraz ścieżkę S łączącą końcowe wierzchołki P_1 i P_2. Ścieżka ta przechodzi przez z. Liczba pokolorowanych wierzchołków na S wynosi:

$$|S \cap C_{\text{OPT}}| = |(P_1 \setminus P_2) \cap C_{\text{OPT}}| + |(P_2 \setminus P_1) \cap C_{\text{OPT}}| = k + k = 2k$$

Jeśli $k \geq 1$, to $2k > k$, co oznacza, że oryginalne kolorowanie C_{OPT} było niepoprawne. To jest sprzeczność. ↯

Zatem na wspólnym odcinku każdej pary zepsutych ścieżek musi znajdować się co najmniej jeden pokolorowany wierzchołek. To implikuje, że istnieje wierzchołek $w \in C_{\text{OPT}}$, który leży na każdej ścieżce ze zbioru $\mathcal{P}_{\text{zepsute}}$. □

Skoro udowodniliśmy istnienie takiego wspólnego wierzchołka w, wybierzmy go tak, aby był jak najbliżej liścia v na ścieżce.

Zdefiniujmy nowe kolorowanie:

$$C'_{\text{OPT}} = (C_{\text{OPT}} \setminus \{w\}) \cup \{v\}$$

Zauważmy, że $|C'_{\text{OPT}}| = |C_{\text{OPT}}|$, więc jeśli jest ono poprawne, to jest również optymalne.

Dowód poprawności C'_{OPT}

Niech S będzie dowolną ścieżką w T.

i. $\boldsymbol{w \notin S}$:
Liczba pokolorowanych wierzchołków na S mogła się co najwyżej zwiększyć o 1 (jeśli $v \in S$). Jeśli by to spowodowało „zepsucie" ścieżki S, oznaczałoby to, że $|S \cap C_{\text{OPT}}|$ = k i $v \in S$. Ale wtedy S należałoby do $\mathcal{P}_{\text{zepsute}}$. Z naszego lematu wiemy jednak, że każda ścieżka z $\mathcal{P}_{\text{zepsute}}$ musi zawierać w. To jest sprzeczność z założeniem $w \notin S$. Zatem ten przypadek jest bezpieczny.

ii. $\boldsymbol{w \in S}$:
Liczba pokolorowanych wierzchołków na S w C'_{OPT} wynosi $|S \cap C_{\text{OPT}}| - 1$ (jeśli $v \notin S$) lub $|S \cap C_{\text{OPT}}|$ (jeśli $v \in S$). W obu sytuacjach liczba ta nie przekracza k, ponieważ $S \cap C_{\text{OPT}} \leq k$. Ten przypadek również jest bezpieczny.

Pokazaliśmy, że możemy zamienić w na v, otrzymując nowe, poprawne i wciąż optymalne kolorowanie C'_{OPT}, które zawiera liść v.

Możemy powtarzać ten proces dla każdego liścia z L_0, który nie należy do C_{OPT}. Po skończonej liczbie kroków przekształcimy C_{OPT} w inne optymalne rozwiązanie C^*_{OPT}, które zawiera wszystkie liście z L_0.

Możemy zastosować to samo rozumowanie do podproblemu $(T - L_0, k - 2)$. Indukcyjnie dochodzimy do wniosku, że istnieje rozwiązanie optymalne, które ma dokładnie taką samą konstrukcję jak C_{ALG}. Zatem $|C_{\text{ALG}}| = |C_{\text{OPT}}|$.

■